# Intelligent Agents for Internet-Based Military Training

Niraj Joshi and V.C. Ramesh
*Illinois Institute of Technology*

*In a project sponsored by the U.S. Army Research Office, we developed and delivered an Internet-based multi-agent system for military training. We describe the client/server-based software architecture of this multiuser intelligent agent (logobot) application that focuses on bookmark automation. We pinpoint some software design issues, such as the inevitable trade-off between real-time response and the complexity of the knowledge base, which will likely be encountered in similar Internet-based military training applications.*

The U.S. military has several training facilities geared toward military personnel who are geographically distributed. Like other military units, the U.S. Army offers mail-based correspondence courses from battle laboratories such as Combined Arms Support Command (CASCOM) in Fort Lee, Va. Although there is no core curriculum, students enroll in different courses based on their military occupational specialty (MOS). The project described in this article is indicative of the steps the Army is taking to migrate from paper-based to electronic training by way of the Internet.

This project provides instructors with tools to gather and categorize Internet uniform resource locators (URLs), and lets students choose categories of URLs that can be automatically stored in their bookmark (favorites) collection. The instructor can exercise control over training materials accessed by students and, as a secondary benefit, can restrict unnecessary "net surfing." Because one of the aims was to provide interdisciplinary training to students, the subject categories for organizing the URLs, which we labeled *domains*, cut across the students' MOS categorization. They can subscribe to any domain they wish, but can only access the URLs provided by the instructor. This effectively constitutes one use of an Intranet wherein the instructor employs the Internet as a source of training material to be housed in an Intranet server that students access.

We devised a software architecture based on the emerging technology of intelligent software agents that we call *logobots* (from *logos* [Greek for *knowledge*] and *bots* [Internet jargon for *intelligent agents*]). This multiagent system has been delivered and installed at CASCOM. This project demonstrates the capabilities added by agent technology to the more traditional client-server software architectures that are commonplace in military applications. We highlight some of the design issues that are likely to be faced by other software engineers who develop Internet-based training applications.

## Logobots

An intelligent agent is a software program that autonomously senses the environment, acts upon it, and over time acquires competence by learning from the environment [1]. Depending on their functionality, agents can be described differently. An agent that acts as a personal assistant is called an *interface agent* in [2]. An agent that has access to at least one and potentially many information sources, and is able to collate and manipulate information obtained from these sources and respond to user queries, is called an *information agent* in [3].

Logobots, which are best described as task-specific autonomous software agents [4], are both interface agents and information agents, as we will elaborate below. One logobot is assigned to each instructor or student, and each augments its knowledge base as new information becomes available. Each logobot maintains training information for its user and has user-interface capabilities to acquire and maintain a knowledge base that pertains to each individual and training domain. We employ two types of logobots: *student logobot* and *instructor logobot*.

The student logobot acts as the student's personalized interface to the remote training. It uses an acquired knowledge base of preferences and reference material and maintains each student's bookmarks. Salient features enabled by this agent include

- *On-demand* or automated update of training material.
- An intuitive interface to the training material, which includes the ability to search the student's bookmarks by specifying domain, URL, title, or description.
- Personalization by selection of training domains and other preferences.
- Ability to change the default preferences assigned by the instructor.

These capabilities provide flexibility for students to identify their training needs. This not only motivates students but also helps them explore the paradigm of *learning to learn* and *just-in-time learning.*

The instructor logobot acts as both an interface agent and an information agent. It helps instructors easily acquire training information from various sources and automatically disseminate it appropriately to the students. It can search the Internet and helps instructors filter relevant information and classify it into different domains. It also assigns default training preferences for different classes of students. Multiple instructor logobots maintain a centralized knowledge base for various training domains. Each instructor logobot

can be personalized. Salient features enabled by this agent are

- An *Internet search interface* based on Java client-server technology, which uses popular search engines like AltaVista to locate training information.
- Filtering of the search results and classification of filtered sites. Information about the state of each document or site is maintained as a part of the central knowledge base; the agent maintains and uses this information to filter out previously seen URLs.
- Facilities for management of domain information, student information to link students to the domains, and for setting the default preferences.

## System Architecture

Logobots are a multiagent system that contains agents and a centralized repository [5]. It is designed to operate in a distributed multiuser (Internet) environment. From a knowledge engineering viewpoint, this architecture could be considered a blackboard architecture [6], which is implemented using client-server tools and techniques. Agents (logobots) act upon the central knowledge base according to user requests and make changes to the repository according to the state of repository data. Logobots have unique identity and built-in security and authentication information. Users need to authenticate before logobots let them access the central knowledge base.

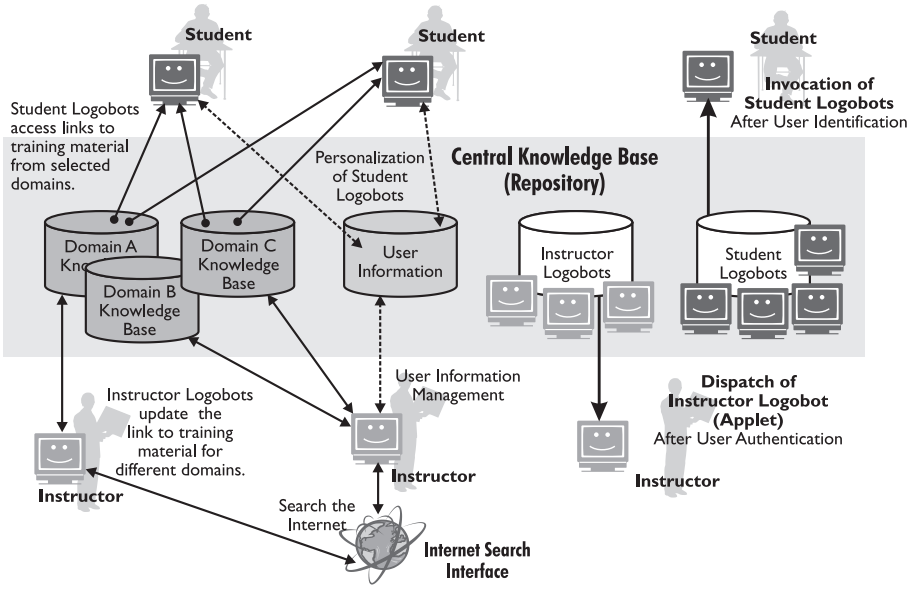
Figure 1 illustrates the system architecture. After proper authentication, students or instructors invoke their logobots. The central knowledge base, which is resident on a networked server with an operational Web server, contains domain information and training material specific to the domains, as well as user or logobot information; the current *state* of the knowledge base, which includes the *accept bin* and *reject bin*, is for memorizing processed information.

Figure 2 displays the internal architecture of the logobots and shows the various interactions between their modules and the central knowledge base. As previously indicated, all logobots are dispatched from a central site (Web server), which houses the central knowledge base and its access modules. Once the logobots are launched into the user's Web browser, they display a personalized initiating page for the user. The user can navigate through the system, accessing various training materials in different domains. Both of the logobots provide a keyword-based search facility over the domain knowledge base. Use of the *search and personalization* mechanism allows students to individually choose their preferences for the sections of training material and to search the material for references.

Instructor logobot supports additional functions to maintain the central knowledge base and manage the user information. Through the *Internet search interface*, the instructor can find training material over the Internet. The *crawler* component contacts various information sources on the Internet and gets the URLs. These URLs are filtered and classified by the *comparator* component, according to the state information in central knowledge base. Exact matches are eliminated, and if necessary, the results are shown to the instructor for further filtering. The instructor may add URLs to certain domains and reject others. These instructor activities are recorded and used later for more efficient search, filtering, and classification.

System knowledge is augmented every time the user performs an action, and the system learns from both the user and the environment. The student and the instructor logobots are client-side components implemented using Java applets, JavaScript, and plain HyperText Markup Language, whereas the central knowledge base is a server-side component implemented with common gateway interface (CGI) programs. The client-server type search modules are implemented using Java applets and applications.

Figure 1. *Overview of the multiagent logobots system.*




## Design Issues

Instructor logobots perform filtering and classification functions based on a learning algorithm, which is a simplified version of a technique called memory-based reasoning (MBR) [3]. The search interface monitors the selection and rejection of bookmarks whenever the instructor is adding training material. Bookmarks and related information are stored in accept and reject bins and later used to filter, classify, and search for new material.

With a little modification, advanced learning techniques like neural nets or semantic nets could augment this simple learning scheme. However, this software had to be usable at CASCOM

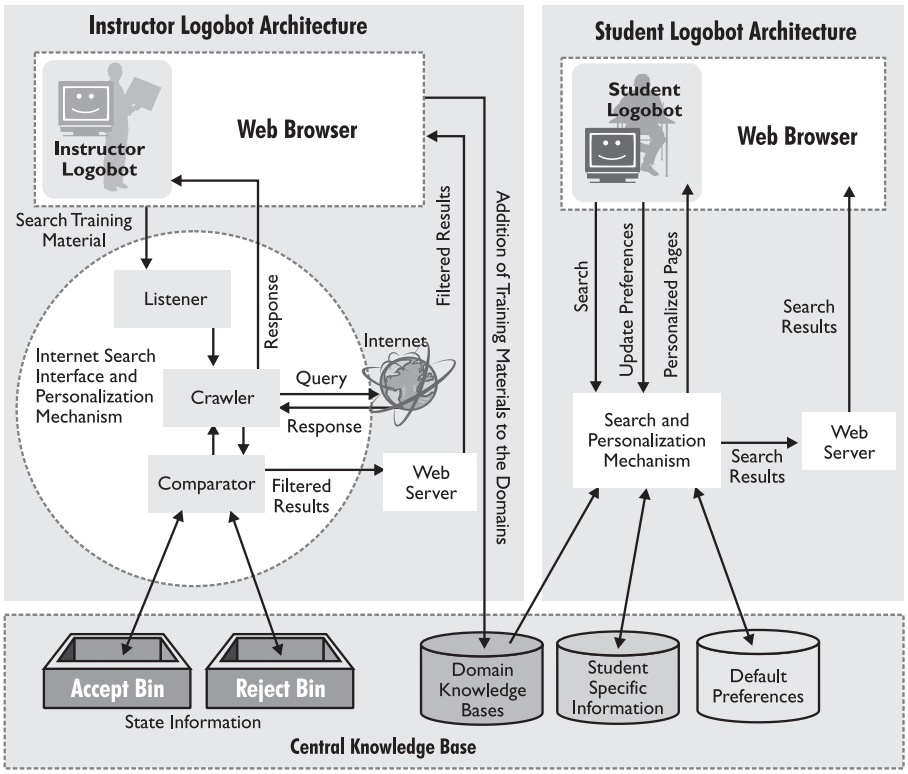



Figure 2. *Internal architecture of logobots.*

as quickly as possible after installation; upon considering the time it takes to train instructors in these advanced schemes, we implemented the simpler MBR scheme. Such trade-offs are often inevitable with intelligent agent-based software systems.

The decision to use the MBR learning algorithm led to another design trade-off: response time vs. the size of the state information stored in the accept and reject bins in the central knowledge base. Recall that for each URL, at a minimum, state information includes the title of the corresponding Web page. URLs in the accept bin also include the domain name. One also could add descriptive information about the contents of the Web page. Clearly, the number of URLs and the complexity of the information stored for each will determine the size of the accept and reject bins. The larger the size of these bins, the better the filtering of the URLs. However, larger accept and reject bins also lead to poorer response times and greater delays, which leads to potential user frustration.

How many URLs should be stored? It is difficult (if not impossible) to determine a generic solution to this design trade-off problem. Based on empirical tests with this implementation of the logobots system, we limited the size of the bins to 100 URLs each. We also limited screening to exact URL matches and excluded titles and descriptions from the filtering process. The trade-off between the size and complexity of knowledge and search response times is a problem that pervades the knowledge engineering field.

A relatively minor but nevertheless important challenge was the need to work across "firewalls." We wanted to minimize use of the inefficient CGI protocol and rely on socket communication primitives provided by Java. However, for Java applets to communicate with the applications running on the central knowledge base server, firewall administrators must provide trusted ports. This was not a problem in this particular military application, but security will play an important role in the design of any such agent system, especially for military applications.

The last design issue concerns why we chose the Internet and Java. When we started creating this system, Java and the Web were unproven technologies. However, these technologies turned out to be advantageous for several reasons. One of the main benefits was the ease with which our Java-based instructor logobots could access popular search engines, such as AltaVista, and build on those results for URL filtering. This prevented the need to create a new search engine and let us concentrate on the higher-level filtering and user interface aspects. We believe that such Internet-based agent applications that leverage the client-server infrastructure of the Web will enable the military to migrate its distance learning facilities to the Internet.

## Conclusion

Intelligent software agents are now migrating toward mainstream software systems. We have described one such system for increasing the efficiency of military training. The system can be particularly valuable in controlling the rapid influx of information in times of crisis (such as a war) by filtering and classifying voluminous event reports into pre-defined subject categories, with students receiving only the portion of information relevant to their line of duty. It also reduces the costs involved in providing military training when compared to the traditional paper or CD-ROM-based techniques.

Though our main focus has been to automate the process of Internet-based military training, the resulting multi-agent framework could be used for any multiuser Internet- or Intranet-based application, which requires controlled dissemination of structured information. The system is not limited to URLs, but is readily extensible to other multimedia objects. Logobots are a step in the military's moves toward virtual classrooms. ◆

### About the Authors

**Niraj Joshi** is a graduate student pursuing a master's degree in computer systems engineering at Illinois Institute of Technology. His research inter-

ests include intelligent agents, software engineering, and database systems. He has three years experience in design and development of software systems based on the Internet and databases. He has a bachelor of engineering degree in computer science from M.S. University in Baroda, India.

Illinois Institute of Technology
Electrical and Computer Engineering Dept.
3301 S. Dearborn Street
Chicago, IL 60616-3793
Voice: 312-567-5074
Fax: 312-567-8976
E-mail: npjoshi@ece.iit.edu
Internet: http://www.ece.iit.edu/~npjoshi

V.C. Ramesh is an assistant professor at the Illinois Institute of Technology. His research interests include intelligent agents, software architectures, and simulation. He has been the principal investigator for several research projects sponsored by various Department of Defense agencies, such as the Army Research Office and the Office of Naval Research. He has published more than 30 technical articles. In 1996, the National Science Foundation named him a recipient of the CAREER award. He holds a doctorate in electrical and computer engineering from Carnegie Mellon University.

Illinois Institute of Technology
Electrical and Computer Engineering Dept.
3301 S. Dearborn Street
Chicago, IL 60616-3793
Voice: 312-567-3765
Fax: 312-567-8976
E-mail: vcr@iit.edu
Internet: http://agents.iit.edu/vcr

## References

1. Genesereth, Michael and Steven Ketchpel, "Software Agents," *Communications of the ACM*, July 1994, pp. 48-53.
2. Maes, Pattie, "Agents That Reduce Work and Information Overload," *Communications of the ACM*, July 1994, pp. 30-40.
3. Wooldridge, Michael and Nicholas Jennings, "Intelligent Agents: Theory and Practice," *Knowledge Engineering Review*, Vol. 10, No. 2, 1995, pp. 115-152.
4. Franklin, Stan and Art Graesser, "Is It an Agent, or Just a Program?" Proceedings of the Third International Workshop on Agent Theories, Architecture, and Languages, Springer-Verlag, 1996.
5. Talukdar, S., V.C. Ramesh, R. Quadrel, and R. Christie, "Multi-Agent Organizations for Real-Time Operations," *Proceedings of the Institute of Electrical and Electronics Engineers*, May 1992.
6. Shaw, Mary and David Garlan, *Software Architecture: Perspectives on an Emerging Discipline*, Prentice-Hall, April 1996.



# Coming Events

### Third IEEE International Conference on Requirements Engineering

Dates: April 6-10, 1998
Location: Marriott Hotel, Colorado Springs, Colo.
Subject: Software requirements engineering-related problems and results, evaluations of promising research and practice.
Sponsors: IEEE Computer Society Technical Council on Software Engineering, Fujitsu, MCI, and INCOSE
Contact: Charlene Rauber-Svitek crs@sei.cmu.edu; Internet: http://www.cs.technion.ac.il/~icre98/

### International Information Technology Quality Conference

Dates: April 13-17, 1998
Location: Orlando, Fla.
Theme: "Providing Proven Solutions for the New Millenium"
Keynote Speakers: Phillip Crosby, Tom DeMarco, William Perry, and Howard Rubin
Sponsor: Quality Assurance Institute
Contact: Voice: 407-363-1111; Fax: 407-363-1112; Internet: http://www.qaiusa.com

### Tenth Annual Software Technology Conference (STC '98)

Dates: April 27 – May 2, 1998
Location: Salt Palace Convention Center, Salt Lake City, Utah
Co-hosts: Ogden Air Logistics Center Commander and the Software Technology Support Center
Contact: See page 30, this issue.

### International Conference on Computer Languages 1998

Dates: May 14-16, 1998
Location: Loyola University, Chicago, Ill.
Sponsors: IEEE Computer Society Technical Committee on Computer Languages, and ACM Special Interest Group on Programming Languages
Contact: http://www.math.luc.edu/iccl98/